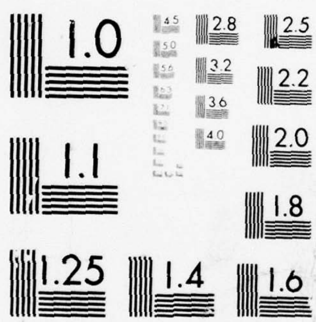
AD-A072 429 YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE F/G 9/2
THEFT AND CONSPIRACY IN THE TAKE-GRANT PROTECTION MODEL.(U)
NOV 78 L SNYDER N00014-75-C-0752

UNCLASSIFIED TR-147 NL

1 OF 1
AD
A072429

END
DATE
FILMED
9-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LUX ET VERITAS

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

12

Theft and Conspiracy in the
Take-Grant Protection Model*

Lawrence Snyder

Technical Report #147, November 1978

*This research was funded in part by National Science Foundation Grant MCS77-12517 and the Office of Naval Research Grant N00014-75-C-0752.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
147 (14) TR-147

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(6) Theft and Conspiracy in the Take-Grant Protection Model

5. TYPE OF REPORT & PERIOD COVERED
(9) Technical rept.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) Lawrence Snyder

8. CONTRACT OR GRANT NUMBER(s)
(15) N-00014-75-C-0752, NSF-MCS-77-12517

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Computer Science
Yale University
New Haven, CT 06520

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
(12) 28p.

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Information Systems Programs
Arlington, VA 22217

12. REPORT DATE
(11) Nov[redacted] 1978

13. NUMBER OF PAGES
23

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution of this report is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Take-Grant Model	conspiracy number
capability	conspiracy graph
conspirator	protection graph

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A user in a protection system that aids in the transfer of authority within the system is called a *conspirator*. (The term is not to connote wrong-doing necessarily, since legitimate sharing requires "conspiracy" in this sense.) Using the Take-Grant Protection Model as an abstraction for a capability-based protection system, necessary and sufficient conditions are developed for determining the number of conspirators required to transfer authority for a given system. The conditions are shown to be testable in polynomial time.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

407 051

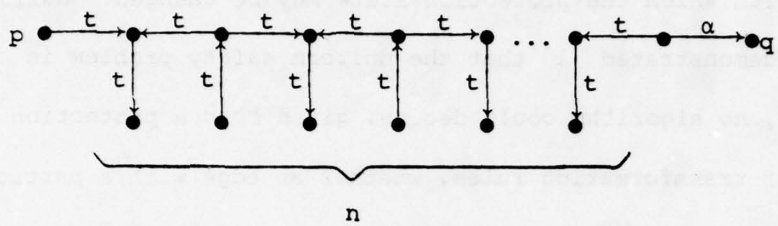
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1. *Introduction*

Models of protection in computer systems usually possess two components, a finite, labeled, directed two color graph representing the protection state of an operating system and a finite set of graph transformation rules with which the protection state may be changed. Harrison, Ruzzo and Ullman demonstrated [1] that the uniform safety problem is undecidable, i.e., no algorithm could decide, given *both* a protection graph and a set of transformation rules, whether an edge with a particular label is ever added to the graph. The Take-Grant Model [2,3,4] has been developed in response to this negative result in order to study such questions for a particular set of transition rules. Linear-time algorithms have been formed for safety-like problems [2,3] for the Take-Grant transition rules. Although the model is simple enough to permit linear time decision procedures, it is rich enough to implement many sharing relationships [4]. In this report we concentrate on the formal development supporting the motivational and interpretive treatments given in [4,5].

First, we characterize the class of graphs that can be created with the Take-Grant rules. Next, the *can·steal* predicate, first introduced in [4] in a limited form, is developed in full generality making it applicable to the common situation of "stealing files." The necessary and sufficient conditions for *can·steal* to be true can still be tested in linear time.

Another main topic is that of quantifying the amount of "cooperation" required to share or steal rights. By the amount of "cooperation" we mean the number of users (i.e., subject vertices) required to initiate rules in order for a particular edge to be added to a graph. This concept was called "conspiracy" in [2] and was studied in [6], where a lower bound is derived. The bound is based on edge incidence and is not tight. For example, the class of graphs of the form



require n+2 conspirators for p to acquire the α edge to q, but in [6] the lower bound for these graphs is 0. The present formulation uses the more flexible notion of "spans" to assess protection graphs. Exact conspiracy measurements for arbitrary protection graphs are derived and an algorithm for discovering minimum conspiracy is presented.

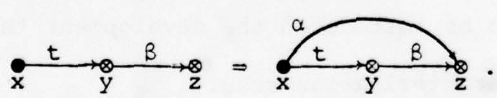
2. *The Take-Grant Model*

The following development of the Take-Grant model follows earlier treatments [2,3,4] and differs in only inessential ways.*

Fix a finite alphabet of labels $R = \{r_1,\ldots,r_m\} \cup \{t,g\}$ called *rights* containing two distinguished elements; "t" is mnemonic of "take" and "g" is mnemonic for "grant." A *protection graph* is a finite, directed, loop-free, two color graph with edges labeled by subsets of R. (Braces around subsets are elided.) Solid vertices, ●, are called *subjects*, empty vertices, ○, are called *objects*; vertices of either type are denoted by ⊗.

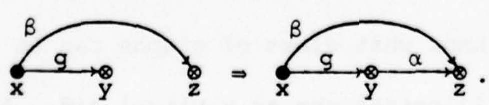
Four rewriting rules are defined to enable a protection graph to change:

Take: Let x, y, and z be three distinct vertices in a protection graph G such that x is a subject. Let there be an edge from x to y labeled γ such that "t" $\in \gamma$, an edge from y to z labeled β and $\alpha \subseteq \beta$. Then the *take* rule defines a new graph G' by adding an edge to the protection graph from x to z labeled α. Graphically,



The rule can be read: "x takes (α to z) from y."

Grant: Let x, y, and z be three distinct vertices in a protection graph G such that x is a subject. Let there be an edge from x to y labeled γ such that "g" $\in \gamma$, an edge from x to z labeled β, and $\alpha \subseteq \beta$. The *grant* rule defines a new graph G' by adding an edge from y to z labeled α. Graphically,



The rule can be read: "x grants (α to z) to y."

----

*Specifically, the "call" rule of [2]has been dropped, r and w (used in [2]), are replaced by t and g, respectively, and "inert" rights [5,6] are permitted.

Create: Let x be any subject vertex in a protection graph G and let α be a subset of R. *Create* defines a new graph G' by adding a new vertex n to the graph and an edge from x to n labeled α. Graphically,

$$\underset{x}{\bullet} \;\Rightarrow\; \underset{x}{\bullet}\xrightarrow{\alpha}\underset{n}{\otimes} \;.$$

The rule can be read: "x creates (α to) new $\left\{\begin{matrix}\text{subject}\\ \text{object}\end{matrix}\right\}$ n.

Remove: Let x and y be any distinct vertices in a protection graph G such that x is a subject. Let there be an edge from x to y labeled β, and let α be any subset of rights. Then *remove* defines a new graph G' by deleting the α labels from β. If β becomes empty as a result, the edge itself is deleted. Graphically,

$$\underset{x}{\bullet}\xrightarrow{\beta}\underset{y}{\otimes} \;\Rightarrow\; \underset{x}{\bullet}\xrightarrow{\beta-\alpha}\underset{y}{\otimes} \;.$$

The rule can be read: "x removes (α to) y."

In these rules, x is called the *initiator*.

Application of rule ρ is denoted by $G \vdash_{\rho} G'$. The reflexive transitive closure of this relation is denoted $G \vdash^{*} G'$. The notation $x \xrightarrow[G]{\alpha} y$ abbreviates "there exists an edge from x to y in G labeled γ and $\alpha \subseteq \gamma$." Figure 1 illustrates* the definitions. Although there are additional concepts to be introduced the development thus far is adequate for proving a characterization result.

3. *Take-Grant Definable Graphs*

In [4] it was argued that the protection graphs actually used in an operating system will be generated by a fixed set of rule protocols, e.g., by the operating system supervisor, editors, compliers, etc. Hence, it is important to know what class of graphs can be generated by

----

*Dashed lines are used in illustrations as a visual aid. Also, even though there is only one directed edge from any vertex a to any vertex b, we occasionally draw two to emphasize changes in labelling.

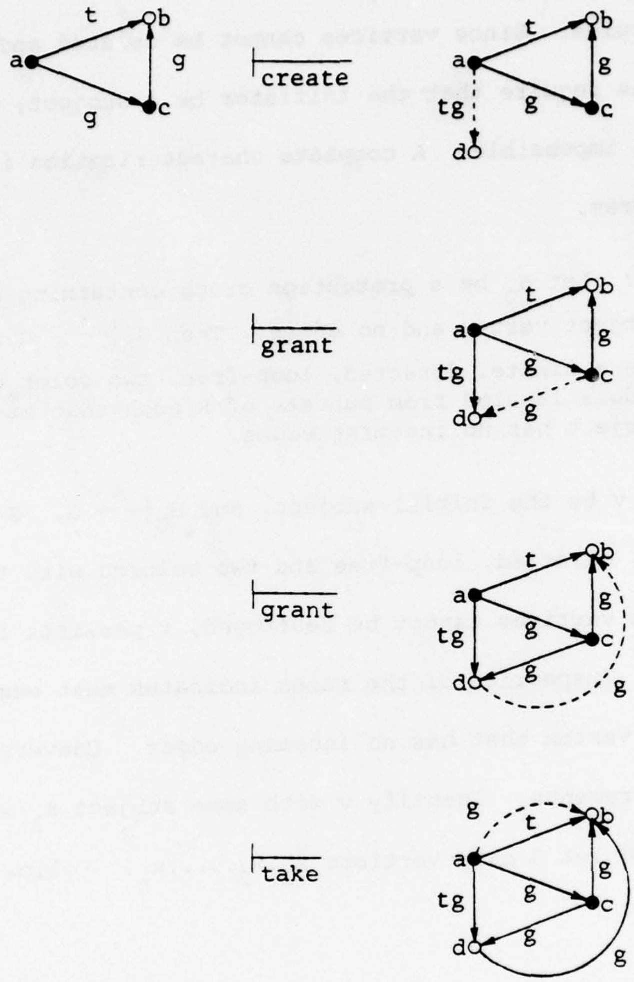


Figure 1: Vertex a acquires g rights to b, i.e., g is added to the label on the a to b edge. The rule applications may be read:

a creates (tg to) new object d,

a grants (g to d) to c,

c grants (g to b) to d,

a takes (g to b) from d.

the Take-Grant rules. Since vertices cannot be deleted and all of the rule applications require that the initiator be a subject, an "all object" graph is impossible. A complete characterization is presented in the next theorem.

Theorem 3.1: Let G_0 be a protection graph containing exactly one subject vertex and no edges. Then $G_0 \overset{*}{\vdash} G$ if and only if G is a finite, directed, loop-free, two color graph with edges labeled from subsets of R such that at least one subject has no incoming edges.

Proof: Let v be the initial subject, and $G_0 \overset{*}{\vdash} G$. G is obviously finite, directed, loop-free and two colored with the indicated labelling. Since vertices cannot be destroyed, v persists in any graph derived from G_0. Inspection of the rules indicates that edges cannot be directed to a vertex that has no incoming edges. Conversely, let G satisfy the requirements. Identify v with some subject x_1 with no incoming edges and let G have vertices $x_1, x_2, \ldots, x_n$. Follow these steps:

(3.1) Perform "v creates ($\alpha \cup \{g\}$ to) new x_i for all x_i ($2 \le i \le n$) where α is the union of all edge labels incoming to x_i in G;

(3.2) For all x_i, x_j such that $x_i \xrightarrow[G]{\alpha} x_j$ perform "v grants (α to x_j) to x_i;"

(3.3) If β is the (possibly empty) set of edges from x_1 to x_i in G, then execute "v removes ($(\alpha \cup \{g\}) - \beta$) to x_i" for $2 \le i \le n$.

The result follows by a simple induction. □

In the next corollary, "component" means connected component.

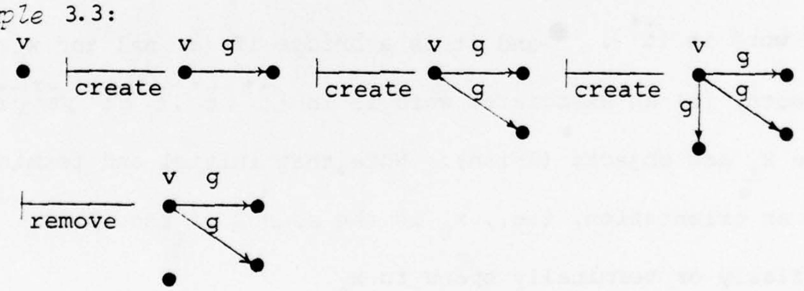
Corollary 3.2: A k component, n edge protection graph can be constructed from a single subject in t rule applications, where $2(k-1)+n \le t \le 2(k-1)+3n$.

Proof: To see the lower limit, note that rules (3.1) and (3.3). are each required for k-1 of the components; the remaining component contains v. Each edge requires at least one application of (3.2). To see the upper limit note that rules (3.1) and (3.3) are sufficient to form one vertex in each component. For each edge charge one application of (3.1) to create its source vertex, one application of (3.2) to assign the edge to the target, and, possibly, one application of (3.3) to delete the edge from v. □

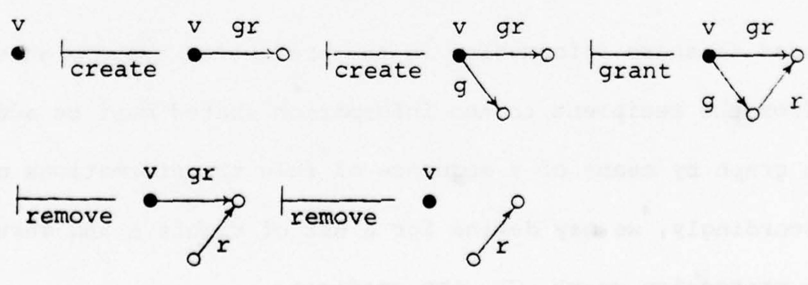
Clearly, the bounds are both achievable as the following example illustrates:

Example 3.3:

(a)



(b)



4. *Predicates and earlier results*

Several properties of paths will be extremely important in our later development. A sequence of vertices $x_0,\ldots,x_n$ is a *path* in G if $x_i \xrightarrow[G]{} x_{i+1}$ or $x_{i+1} \xrightarrow[G]{} x_i$, $0 \le i < n$. Thus paths are defined independent of direction. Vertices p and q of G are *tg-connected* if there is a path $p = x_0,\ldots,x_n = q$ and the label α on the edge between x_i and x_{i+1} contains t or g. An *island* of G is a maximal, tg-connected subject-only subgraph of G.

The *edge alphabet* is composed of four letters $\{\vec{t},\vec{g},\overleftarrow{t},\overleftarrow{g}\}$. Let $x \xrightarrow[G]{t} y$ (resp. $x \xrightarrow[G]{g} y$) then the letter $\vec{t}$ (resp. $\vec{g}$) is *associated* with the edge. Words are associated with paths in the obvious way; for example, $\bullet \xrightarrow{t} \bullet \xleftarrow{tg} \bullet \xleftarrow{g} \bullet$ has the words $\vec{t}\overleftarrow{t}\overleftarrow{g}$ and $\vec{t}\overleftarrow{g}\overleftarrow{g}$ associated with it. A path $x_0,\ldots,x_n$ is an *initial span* if it has an associated word in $\{\vec{t}^{\,*}\vec{g}\}$, it is a *terminal span* if $n>0$ and it has an associated word in $\{\vec{t}^{\,*}\}$, and it is a *bridge* if (a) $n>1$ and x_0 and x_n are subjects, (b) an associated word is in $\{\vec{t}^{\,*}, \overleftarrow{t}^{\,*}, \vec{t}^{\,*}\vec{g}\overleftarrow{t}^{\,*}, \vec{t}^{\,*}\overleftarrow{g}\overleftarrow{t}^{\,*}\}$, and (c) the x_i are objects $(1<i<n)$. Note that initial and terminal spans have an orientation, i.e., x_0 is the *source* of the spans. We say x_0 initially or terminally spans to x_n.

In order to share information in the protection system, an edge pointing from the recipient to the information shared must be added to the protection graph by means of a sequence of rule transformations of the graph. Accordingly, we may define for a set of rights α and vertices p and q of a protection graph G_0, the predicate

can•share$(\alpha,p,q,G_0) \Leftrightarrow$ there are protection graphs $G_1,\ldots,G_n$ such that $G_0 \vdash^{*} G_n$ and $p \xrightarrow[G_n]{\alpha} q$.

When interest is restricted to protection graphs containing only subjects, we have

Theorem 4.1 [2]: For a subject only protection graph G_0, can•share(α,p,q,G_0) is true if and only if the following two conditions hold.

Condition 1: There exist vertices $s_1,\ldots,s_u$ such that for each i, $1 \le i \le u$; $s_i \xrightarrow[G_0]{\gamma_i} q$ and $\alpha = \gamma_1 \cup \ldots \cup \gamma_u$;

Condition 2: p is tg-connected to each s_i, $1 \le i \le u$.

The conditions under which *can•share* holds for general protection graphs are somewhat more complicated. In particular, Condition 1 must be augmented by Condition 3:

Condition 3: There exist subject vertices p' and $s_1',\ldots,s_t'$ such that

(a) $p = p'$ or p' initially spans to p;
(b) $s_i = s_i'$ or s_i' terminally spans to s_i;

and Condition 2 must be recast in terms of bridges and islands:

Condition 4: For each (p',s_i') pair $(1 \le i \le u)$ there exist islands $I_1,\ldots,I_v$ $(v \ge 1)$ such that $p' \in I_1$, $s_i' \in I_v$ and there is a bridge from I_j to I_{j+1} $(1 \le j < v)$.

Clearly, Condition 4 is simply Condition 2 for the case v = 1. The counter part to Theorem 4.1 for general protection graphs is

Theorem 4.2 [3]: The predicate *can•share*(α,p,q,G_0) is true if and only if Conditions 1, 3, and 4 hold.

As corollaries, it is known that there are algorithms operating in linear time in the size (V+E) of the graph to test both predicates.

5. *Theft*

The *can•share* predicate presumes perfect cooperation from all users (i.e., subjects). The *can•steal* predicate must capture the notion that a subject vertex acquires a new right without any cooperation from an original owner. Formally, for two vertices p and q in a protection graph G_0, and right α, define

can•steal$(\alpha,p,q,G_0) \Leftrightarrow \sim p \xrightarrow[G_0]{\alpha} q$ and there exist protection graphs $G_1,\ldots, G_n$ such that

(5.1) $G_0 \vdash_{\rho_1} G_1 \vdash_{\rho_2} \ldots \vdash_{\rho_n} G_n$;

(5.2) $p \xrightarrow[G_n]{\alpha} q$, and

(5.3) if $s \xrightarrow[G_0]{\alpha} q$ then no ρ_j has the form "s grants (α to q) to x_i" for any $x_i \in G_{j-1}, 1 \le j \le n$.

Clearly, p, q and s must be distinct since these are protection graphs.

Theorem 5.1: For vertices p and q in a protection graph, G_0 and right α, *can•steal*(α,p,q,G_0) if and only if the conjunction of the following conditions holds:

(i) $\sim p \xrightarrow[G_0]{\alpha} q$,

(ii) there is a subject p' such that p = p' or p' initially spans to p,

(iii) there is a vertex s such that $s \xrightarrow[G_0]{\alpha} q$ and *can•share*(t,p,s,G_0).

Proof: ($\Rightarrow$) Suppose *can•steal*(α,p,q,G_0) is true. Condition (i) of the theorem holds by definition. Let n be the smallest integer such that $G_0 \vdash_{\rho_1} G_1 \vdash_{\rho_2} \ldots \vdash_{\rho_n} G_n$ and $p \xrightarrow[G_n]{\alpha} q$. If p is a subject, (ii) holds, so suppose p is an object. If no p' exists, then for all x *can•share*(α,p,x,G_0) is false, contradicting (4.2). Similar reasoning assures the existence of x such that $s \xrightarrow[G_0]{\alpha} q$, so we concentrate on showing the necessity of *can•share*(t,p,s,G_0). Let $T = \{s | s \xrightarrow[G_0]{\alpha} q\}$. Let i be the least index such that in G_i there is a vertex z_1, and $z_1 \xrightarrow[G_i]{\alpha} q$, but $\sim z_1 \xrightarrow[G_{i-1}]{\alpha} q$. The operation causing this edge to be added cannot be a grant, since *can•steal* is true and those vertices pointing to q with α labels in G_{i-1} are the same as those in G_0. The operation must be a take of the form:

$$\begin{tikzcd} \underset{z_1}{\bullet} \arrow[r, "t"] & \underset{s}{\otimes} \arrow[r, "\alpha"] & \underset{q}{\otimes} \end{tikzcd} \Rightarrow \begin{tikzcd} \underset{z_1}{\bullet} \arrow[r, "t"] \arrow[rr, bend left, "\alpha"] & \underset{s}{\otimes} \arrow[r, "\alpha"] & \underset{q}{\otimes} \end{tikzcd}$$

for some $s \in T$. Let $z_2, \ldots, z_\ell = p$ be the other vertices (in order of appearance) that are assigned α labeled edges to q in the derivation. Then an alternative derivation could be formed where each rule of the form

z_j takes (α to q) from x_j

or

x_j grants (α to q) to z_j

is replaced by

z_j takes (t to s) from x_j

or

x_j grants (t to s) to z_j,

respectively, for $2 \le j \le \ell$, provided $x_j = z_{j-1}$. But this latter equality most hold since the derivation is a shortest one. Thus, *can•share*(t,p,s,G_0) proving that (iii) holds.

(⇐) Suppose the three conditions hold. Then if p is a subject, the theorem is immediately satisfied since p can take (α to q) from s once it gets the t right to s. If p is an object then *can•share*(t,q,s,G_0) implies there is some subject p' initially spanning to p and *can•share*(t,p',s,G_0). If $\sim p' \xrightarrow[G_0]{\alpha} q$ then p' can take the right (α to q) from s and grant it to p. If $p' \xrightarrow[G_0]{\alpha} q$ then the following sequence enables p' to form a surrogate vertex n to transmit the right (α to q) to p given that $p' \xrightarrow[G_0]{t} s$ and $p' \xrightarrow[G_i]{g} p$:

p' creates (g to) a new subject n;

p' grants (t to s) to n;

p' grants (g to p) to n.

(These steps are legal even if α=t.)
Then n completes the task with operations:

n takes (α to q) from s;

n grants (α to q) to p.

This is a witness for *can•steal*(α,p,q,G_0) proving the theorem. □

Corollary 5.2: There is an algorithm to test the *can•steal* predicate that operates in time linear in the size of the protection graph.

6. *Conspiracy*

In this section we are concerned with the amount of "cooperation" required to effect the sharing or stealing. This cooperation has been called "conspiracy" [2] and for a given sequence of legal rule applications $\rho_1,\ldots,\rho_n$, it is simply $|\{x|x \text{ initiates } \rho_i\}|$. Our concern in this section is determining for a given true predicate *can•share*(α,p,q,G_0) the minimum conspiracy required to produce a G_n that is a witness to its truth. We will be able to find the exact value for arbitrary protection graphs.

Let G be a protection graph and y a subject vertex, then the *access-set with focus y*

$A(y) =_{def} \{y\} \cup \{x|y$ initially spans or terminally spans to $x\}$.

Clearly, for a given focus y in G, A(y) in unique. Access sets will be used to measure the size of the conspiracy.

For the remainder of the section, we restrict our attention to protection graph G with vertices $p = x_0,\ldots,x_n = s$, $x_{n+1} = q$. An edge in G either forms a tg-connection between x_{i-1} and x_i ($1 \le i \le n$) or is $s \xrightarrow{\alpha} q$. We suppose that *can•share*(α,p,q,G) holds.

Say that a vertex is a *tg-sink* if

(6.1) the vertex is x_0 and the only letter associated with the x_0,x_1 edge is $\overleftarrow{t}$,

(6.2) the vertex has incident edges whose only associated word is in $\{\overrightarrow{t}\overleftarrow{t},\overrightarrow{g}\overleftarrow{g}\}$ or

(6.3) the vertex is x_n and the only letter associated with the x_{n-1},x_n edge is $\overrightarrow{g}$.

The motivation for this definition will become evident in the claim of Theorem 6.1.

An *access-set cover for G with foci $y_1,\ldots,y_u$* is a family of sets $A(y_1),\ldots,A(y_u)$ such that for each i ($1 \le i \le n$) vertices $\{x_{i-1},x_i\} \subseteq A(y_j)$ for some j, $1 \le j \le u$. Note that the subject requirement of access-sets might prevent certain tg-connected paths from having a cover. It will become clear from the subsequent theorems, however, that a tg-path has an access-set cover if and only if *can·share*(α,p,q,G_0) is true. Finally, an access set cover is said to be *minimal* if it minimizes u over all access set covers.

First we establish a lower bound.

Theorem 6.1: Let G_0 be a tg-connected path $p = x_0,\ldots,x_n = s$ such that *can·share*(α,p,q,G_0) is true. Let k be the number of access sets in a minimal cover of G_0, and ℓ the number of tg-sinks. Then $k+\ell$ initiators are necessary.

Proof: Let $\rho_1,\ldots,\rho_v$ be the minimal set of rules required for a minimal set of initiators $y_1,\ldots,y_u$ to implement *can·share*(α,p,q,G_0). To see that the access sets $A(y_1),\ldots,A(y_u)$ with initiator foci $y_1,\ldots,y_u$ cover G_0, note that $x \notin A(y_i)$ for all i implies that no initiator can take from or grant to x, so x and its incident edges can be removed without affecting rules $\rho_1,\ldots,\rho_v$. But this violates the connectedness Condition 4 of *can·share*. Thus, the access sets $A(y_1),\ldots,A(y_u)$ at least cover G_0.

Claim: Every vertex x_i that is a tg-sink must be an initiator.

Proof of Claim: First note that each such x_i must be a subject by Condition 4. Suppose x_i fails to satisfy the claim and $\overrightarrow{t}\overleftarrow{t}$ is associated with x_i's incident edges. Then no rule ρ_j of the form "z takes (β to y) from x_i" is ever executed since x_i has no out edges and it cannot be assigned any. Furthermore, since v, the number of rules, is minimal, no rules of the form "z takes (t to x_i) from x_{i-1}" or "x_{i-1} grants (t to x_i) to z" are ever executed since no use could be made of the t right thus assigned; a similar situation holds for x_{i+1} transmitting its t right to x_i. Thus x_i and its incident edges can be deleted violating the connectedness Condition 4.

If $\overleftrightarrow{gg}$ is associated with x_i's incident edges, no rule ρ_j of the form "z grants (β to y) to x_i" is ever executed since that right cannot be transmitted by x_i and v is assumed minimal. As with the $\overleftrightarrow{tt}$ case there is no need for any ρ_j to transmit the g right, so x_i can be eliminated and thus the connectedness condition is violated. The situation for the end points is analogous. The claim follows.

Let $y_1,\ldots,y_\ell$ be the tg-sink initiators. Then $A(y_1),\ldots,A(y_\ell)$ are singleton sets. Moreover, each of these vertices is a member of its adjacent access-sets. Thus, the other access-sets, $A(y_{\ell+1}),\ldots,A(y_{\ell+k})$ ($\ell+k = u$) constitute a cover for G_0. The theorem follows. □

Some discussion is in order. Basically, edges can be transmitted by an initiator to any vertex in its access set. Edges are passed "along the path" because access sets will overlap. If one initiator can take from the common element and the other can grant to it, then edges can move from one access set to the next. But if the common vertex is a tg-sink, then it must aid in the communication.

Next we establish a matching upper bound, but first a lemma will simplify matters.

Lemma 6.2: Let $x_0,\ldots,x_n$ be a tg-connected path and $A(y_1),\ldots,A(y_k)$ a minimal access-set cover ordered by increasing indices of x_i. If $y_{i+1} \xrightarrow[G]{\alpha} q$ then there exists G' such that $y_i \xrightarrow[G']{\alpha} q$ and all rules in $G \vdash^{*} G'$ are initiated by y_i, y_{i+1}, and perhaps, their common element.

Proof: Let $z = A(y_i) \cap A(y_{i+1})$. Consider the spans to z from y_i and y_{i+1}. The notation "take* r" means "perform enough takes to acquire" right r.

	span from y_i to z	span from y_{i+1} to z	rule sequence
(6.4)	terminal($\overrightarrow{t}^*$)	terminal($\overleftarrow{t}^*$)	z is necessarily a subject, since $\overrightarrow{t}^*\overleftarrow{t}^*$ isn't a bridge. (a) z creates (tg to) new n, (b) y_{i+1} takes* (g to n) from z via elements of the span, (c) y_{i+1} grants (α to q) to n (d) y_i takes* (α to q) from n.
(6.5)	terminal($\overrightarrow{t}^*$)	initial($\overleftarrow{g}\overleftarrow{t}^*$)	(a) y_{i+1} takes* (g to z) from elements of the span, (b) y_{i+1} grants (α to q) to z, (c) y_i takes (α to q) from z.
(6.6)	initial($\overrightarrow{t}^*\overrightarrow{g}$)	terminal($\overleftarrow{t}^*$)	(a) y_i creates (tg to) new n, (b) y_i takes* (g to z) from elements of the span, (c) y_i grants (g to n) to z, (d) y_{i+1} takes* (g to n) from z via elements of the span, (e) y_{i+1} grants (α to q) to n, (f) y_i takes (α to q) from n.
(6.7)	initial($\overrightarrow{t}^*\overrightarrow{g}$)	initial($\overleftarrow{g}\overleftarrow{t}^*$)	z is necessarily a subject since $\overrightarrow{t}^*\overrightarrow{g}\overleftarrow{g}\overleftarrow{t}^*$ isn't a bridge. (a) y_i creates (tg to) new n, (b) y_i takes* (g to z) from elements of span, (c) y_i grants (g to n) to z, (d) y_{i+1} grants (α to q) to z via elements of span, (e) z grants (α to q) to n, (f) y_i takes (α to q) from n.

Except for (6.4a) and (6.7e) the vertices initiating the rules are either y_i or y_{i+1}. □

Corollary 6.3: For adjacent access sets $A(y_i)$ and $A(y_{i+1})$, α rights to q can be transferred from y_{i+1} to y_i with no other initiators unless there are consecutive edges labeled $\overrightarrow{t}\overleftarrow{t}$ or $\overrightarrow{g}\overleftarrow{g}$. In this case, one additional operation initiated by $z = A(y_i) \cap A(y_{i+1})$ is sufficient.

Let *can•share*(α,p,q,G_0) hold via the tg-connected path $p = x_0,\ldots,x_n = s$ and let $A(y_1),\ldots,A(y_k)$ be a minimal access-set cover. Let ℓ be the number of tg-sinks.

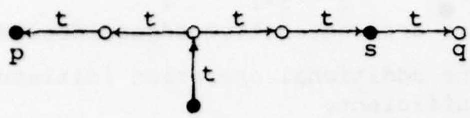
Theorem 6.4: For p to acquire α rights to q, $k+\ell$ initiators suffice.

Proof: Clearly, $p \in A(y_1)$, $s \in A(y_k)$. If $s = y_k$ then $y_k \xrightarrow[G_0]{\alpha} q$. If y_k terminally spans to s, then y_k takes* (α to q) from s via elements of span. If y_k initially spans to s, then s is necessarily a subject by conditions of *can•share* and rules (6.5a-b) (with $s = y_{i+1}$ and $y_k = z$) suffice to transfer (α to q) to y_k. In all three cases $y_k \xrightarrow{\alpha} q$ and we have a basis step. Lemma 6.2 can now be inductively applied, and $y_1 \xrightarrow{\alpha} q$. If $y_1 = p$ we are done. If y_1 initially spans to p then y_1 takes* (g to p) from elements of the span and it grants (α to q) to p. If y_1 terminally spans to p then p is necessarily a subject by conditions on *can•share* and (6.4a-c) (with $p = z$, $i = 0$) suffice to transfer (α to q) to p. (Note, use of (6.4a) implies the addition of another initiator, namely p, but this is counted in the definition of tg-sink. The case is similar for use of (6.5a-b) by above.) □

7. *Conspiracy in general graphs*

Although the theorems of the last section give an exact measurement of the number of initiators required for sharing, they only apply to paths.

In general, extending these results to graphs cannot be done simply by looking for vertex disjoint paths. For example, if G is the graph



the (only) vertex disjoint path from p to s does not qualify as a legal path for *can•share*(α,p,q,G) to hold, even though the predicate is true. Working from the earlier development we now present a finer analysis applicable to general graphs.

Recall that if $v \in A(x)$, the access set with focus x, there are three possible conditions any subset of which v can satisfy: v is the focus of A(x) (i.e., v = x), x initially spans to v or x terminally spans to v. Each of these properties is said to be a *reason* for $v \in A(x)$.

Given a protection graph G with subject vertices $x_1,\ldots,x_n$, we will define a new graph, the *conspiracy graph*, H, determined by G. H has vertices $y_1,\ldots,y_n$ and each y_i has associated with it the access-set $A(x_i)$. There is an undirected edge between y_i and y_j provided $\delta(x_i,x_j) \neq \emptyset$ where δ is called the deletion operation and is defined by:

> $\delta(x,x')$ ⇐ return all elements in $A(x) \cap A(x')$ except those z for which either (a) the only reason $z \in A(x)$ is x initially spans to z and the only reason $z \in A(x')$ is x' initially spans to z or (b) the only reason $z \in A(x)$ is x terminally spans to z and the only reason $z \in A(x')$ is x' terminally spans to z.

The graph thus constructed is called H. See the example in Figure 2.

Let H be constructed from G as just described. Define the sets

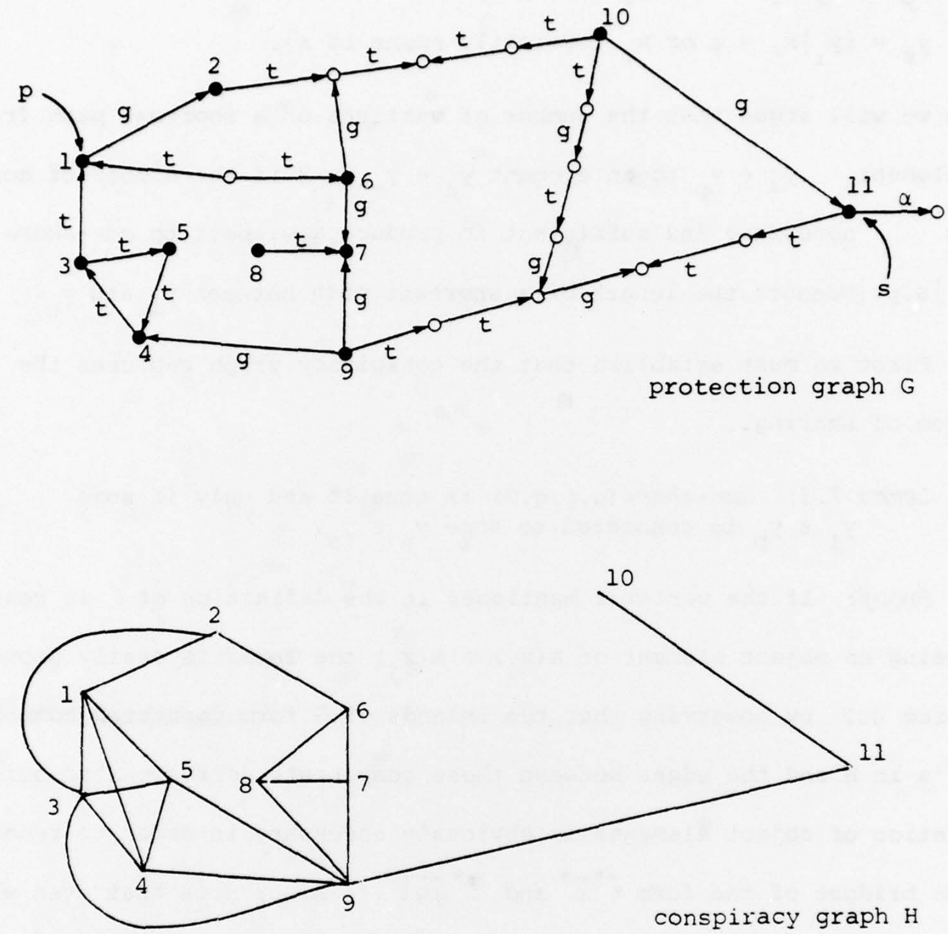


Figure 2: A protection graph and its induced conspiracy graph.

$y_p = \{y_i | x_i = p \text{ or } x_i \text{ initially spans to } p\}$,

$y_s = \{y_i | x_i = s \text{ or } x_i \text{ terminally spans to } s\}$.

Then we will argue that the number of vertices on a shortest path from an element $y_1 \in y_p$ to an element $y_n \in y_s$ in H is the number of conspirators necessary and sufficient to produce a witness to *can·share*(α,p,q,G). Let |s.p.| denote the length of a shortest path between y_1 and y_n.

First we must establish that the conspiracy graph captures the notion of sharing.

Lemma 7.1: *Can·share*(α,p,q,G) is true if and only if some $y_1 \in y_p$ is connected to some $y_n \in y_s$.

Proof: If the vertex z mentioned in the definition of δ is restricted to being an object element of $A(x_i) \cap A(x_j)$ the lemma is easily proved from Theorem 4.2 by observing that the islands of G form connected components of y's in H and the edges between these components correspond to bridges. (Deletion of object elements is obviously necessary in order to remove false bridges of the form $\overrightarrow{t}^{*}\overleftarrow{t}^{*}$ and $\overrightarrow{t}^{*}\overrightarrow{g}\overleftarrow{g}\overleftarrow{t}^{*}$.) Also, note that even with subject deletions, if y_1 and y_n are connected *can·share*(α,p,q,G) is true. So the remaining case is when *can·share*(α,p,q,G) is true but removal (by δ) of z from $A(x_i) \cap A(x_j)$ prevents y_1 and y_n from being connected. Let z be associated with y_z. Note that since z is a focus it has reason to be in $A(x_i) \cap A(z)$ and in $A(z) \cap A(x_j)$. Thus there are edges in H between y_i and y_z and between y_z and y_j. Thus, the absence of an edge between y_i and y_j cannot prevent y_1 and y_n from being connected, since there is a path between y_i and y_j in any case. □

Notice from the proof that the effect of deleting subjects via δ is to prevent two foci, y_i and y_j from being directly connected when their only connecting spans contain a tg-sink. By deleting such vertices, we force y_i and y_j to be connected by a path of two edges -- a means of easily counting the tg-sink as a conspirator.

Theorem 7.2: To produce a witness to *can•share*(α,p,q,G) $|s.p.|$ conspirators are sufficient.

Proof: A simple induction on the spans corresponding to the edges of the s.p. using Lemma 6.2 proves the result provided we observe the following point. Since p,q,s are distinct and the y_i on the s.p. are distinct, all rules given in Lemma 6.2 can be performed provided the foci of the access-sets are different from their common element(s). By inspection of the rules of Lemma 6.2, whenever a focus and common element coincide the rule whose application is prevented (by distinctness of vertices for rule applications, Sec. 2) provides a right that is already possessed (e.g., rule 6.5c, $y_i = z$) or it provides a right used in the subsequent rule to acquire a right already possessed (e.g., rule 6.5a and 6.5b, $y_{i+1} = z$). In these cases the rule whose application is prevented is not needed. □

Theorem 7.3: To produce a witness to *can•share*(α,p,q,G) $|s.p.|$ conspirators are necessary.

Proof: Let $y_1 = z_1,\ldots,z_u = y_n$ be vertices along a shortest path from y_1 to y_n. If there exist only vertex disjoint tg-connected paths in G from z_i to z_{i+1} $(1\le i<u)$ then the z_i are foci of an access-set cover for the path. By construction there are no tg-sinks and if y_1 not associated with p (resp. y_n not associated with s) then the subject associated with y_1 (y_n) initially (terminally) spans to p (s) and so it need not conspire. By theorem 6.1, u conspirators are necessary.

The remaining case is for an induced path that is not vertex disjoint. Although redundant rule applications may arise, it is clear that duplicated vertices along a span are not harmful to the lemma unless they reduce the number of required conspirators. Suppose that conspirators $z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_u$ can produce a witness. Then there is a $w \in A(z_{i-1}) \cap A(z_{i+1})$. But by choice of the z_i vertices on a shortest path there is no edge between z_{i-1} and z_{i+1}. Thus, $w \neq z_{i-1}$, $w \neq z_{i+1}$ and $w \notin \delta(z_{i-1},z_{i+1})$. But this implies (if w is an object) that there is no bridge between z_{i-1} and z_{i+1} (contradicting by Lemma 7.1 the assumption $z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n$ are sufficient) or it implies (if w is a subject) the presence of a tg-sink. By Theorem 6.1 w must be counted as a conspirator. □

8. *Concluding Remarks*

The development of the conspiracy results provides a reasonably clear picture of how sharing is accomplished in the Take-Grant Model. In particular, the notion of access-set describes that portion of a protection graph under direct "control" of the subject which is its focus. Communication outside of this region of influence requires the cooperation of other subjects. This information will doubtless be useful for designers of specific protection systems as explained in [4].

Several problems remain open. First, there is the question of algorithmic complexity of determining the minimum number of conspirators required for a right to be shared. In Section 7 this is determined by finding a shortest path in a conspiracy graph. That question is obviously a linear time process, but the construction of a conspiracy graph (as described) requires n^2 operations for an n subject graph just to fill in the edges. A simpler scheme that does not depend on the explicit construction of the conspiracy graph could be envisaged.

Another issue is to determine for a given graph what set of conspirators must have participated in the sharing of a right after the fact. The test is complicated by the fact that certain rights could have been removed in order to hide the conspiracy. One might be able to infer from the structure of the graph that even though a subject has deleted the conspiratorial rights, they once existed.

Acknowledgement: Thanks are due to Matthew Bishop for useful conversations concerning conspiracy and Michele Boucher for expert preparation of the manuscript.

9. *References*

[1] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman.
Protection in Operating Systems.
CACM, 19,8 (1976).

[2] R. J. Lipton and L. Snyder.
A Linear Time Algorithm for Deciding Subject Security.
JACM, 24:3, pp. 455-464, (1977).

[3] A. K. Jones, R. J. Lipton, and L. Snyder.
A Linear Time Algorithm for Deciding Security.
Proc. 17th FOCS, (1976).

[4] L. Snyder.
Analysis and Synthesis in the Take-Grant System.
Proc. 6th SOSP, (1977).

[5] L. Snyder.
Formal Models of Capability-Based Protection Systems.
Yale Department of Computer Science Technical Report #151, 1978.

[6] T. Budd and R. J. Lipton.
Inert Rights and Conspirators in the Take/Grant System.
Yale Department of Computer Science Technical Report #126, (1977).

OFFICIAL DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, VA 22314	12 copies
Office of Naval Research Arlington, VA 22217	
Information Systems Program (437)	2 copies
Code 200	1 copy
Code 455	1 copy
Code 458	1 copy
Office of Naval Research Branch Office, Boston Bldg 114, Section D 666 Summer Street Boston, MA 02210	1 copy
Office of Naval Research Branch Office, Chicago 536 South Clark Street Chicago, IL 60605	1 copy
Office of Naval Research Branch Office, Pasadena 1030 East Green Street Pasadena, CA 91106	1 copy
Naval Research Laboratory Technical Information Division, Code 2627 Washington, D.C. 20375	6 copies
Dr. A. L. Slafkosky Scientific Advisor Commandant of the Marine Corps (Code RD-1) Washington, D.C. 20380	1 copy
Naval Ocean Systems Center Advanced Software Technology Division Code 5200 San Diego, CA 92152	1 copy
Mr. E. H. Gleissner Naval Ship Research and Development Center Computation and Mathematics Department Bethesda, MD 20084	1 copy

Captain Grace M. Hopper (008) Naval Data Automation Command Washington Navy Yard Building 166 Washington, D.C. 20374	1 copy
Defense Advanced Research Projects Agency Attn: Program Management/MIS 1400 Wilson Boulevard Arlington, VA 22209	3 copies